

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

IN RE APPLICATION OF: Mitsunobu YOSHIDA, et al.

GAU:

SERIAL NO: NEW APPLICATION

EXAMINER:

FILED: HERewith

FOR: SIMULATION METHOD AND PROGRAM PRODUCT

REQUEST FOR PRIORITY

COMMISSIONER FOR PATENTS
ALEXANDRIA, VIRGINIA 22313

SIR:

- ☐ Full benefit of the filing date of U.S. Application Serial Number _____, filed _____, is claimed pursuant to the provisions of 35 U.S.C. §120.
- ☐ Full benefit of the filing date(s) of U.S. Provisional Application(s) is claimed pursuant to the provisions of 35 U.S.C. §119(e):
Application No. _____ Date Filed _____

- ☒ Applicants claim any right to priority from any earlier filed applications to which they may be entitled pursuant to the provisions of 35 U.S.C. §119, as noted below.

In the matter of the above-identified application for patent, notice is hereby given that the applicants claim as priority:

<u>COUNTRY</u>	<u>APPLICATION NUMBER</u>	<u>MONTH/DAY/YEAR</u>
Japan	2002-376210	December 26, 2002

Certified copies of the corresponding Convention Application(s)

- ☒ are submitted herewith
- ☐ will be submitted prior to payment of the Final Fee
- ☐ were filed in prior application Serial No. _____ filed _____
- ☐ were submitted to the International Bureau in PCT Application Number _____
Receipt of the certified copies by the International Bureau in a timely manner under PCT Rule 17.1(a) has been acknowledged as evidenced by the attached PCT/IB/304.
- ☐ (A) Application Serial No.(s) were filed in prior application Serial No. _____ filed _____; and
- ☐ (B) Application Serial No.(s) _____
- ☐ are submitted herewith
- ☐ will be submitted prior to payment of the Final Fee

Respectfully Submitted,

OBLON, SPIVAK, McCLELLAND,
MAIER & NEUSTADT, P.C.


Marvin J. Spivak

Registration No. 24,913

Customer Number

22850

Tel. (703) 413-3000
Fax. (703) 413-2220
(OSMMN 05/03)

C. Irvin McClelland
Registration Number 21,124

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日 2 0 0 2 年 1 2 月 2 6 日
Date of Application:

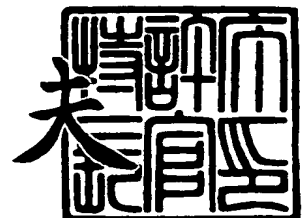
出 願 番 号 特 願 2 0 0 2 - 3 7 6 2 1 0
Application Number:
[ST. 10/C]: [J P 2 0 0 2 - 3 7 6 2 1 0]

出 願 人 株 式 会 社 東 芝
Applicant(s):

2 0 0 3 年 7 月 1 8 日

特許庁長官
Commissioner,
Japan Patent Office

今 井 康 夫



【書類名】 特許願

【整理番号】 13B02Y0591

【あて先】 特許庁長官殿

【国際特許分類】 G06F 15/00

【発明の名称】 シミュレーション方法およびシミュレーションプログラム

【請求項の数】 4

【発明者】

【住所又は居所】 神奈川県川崎市幸区小向東芝町 1 番地 株式会社東芝
研究開発センター内

【氏名】 吉田 充伸

【発明者】

【住所又は居所】 神奈川県川崎市幸区小向東芝町 1 番地 株式会社東芝
研究開発センター内

【氏名】 近藤 浩一

【特許出願人】

【識別番号】 000003078

【氏名又は名称】 株式会社 東芝

【代理人】

【識別番号】 100083161

【弁理士】

【氏名又は名称】 外川 英明

【電話番号】 (03)3457-2512

【手数料の表示】

【予納台帳番号】 010261

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1



【物件名】

要約書 1

【プルーフの要否】

要



【書類名】 明細書

【発明の名称】 シミュレーション方法およびシミュレーションプログラム

【特許請求の範囲】

【請求項 1】 ハイブリッドモデルを利用した記述データに基づいて、対象とする機構の挙動を時間軸に沿ってシミュレーションするシミュレーション方法であって、

前記記述データに含まれる連続系方程式の記述、状態遷移に伴う該連続系方程式の切り替えに関する記述、及び前記連続系方程式で表現される処理以外の処理に関する記述を抽出するため、前記記述データを解析する解析ステップと、

前記解析ステップにおいて抽出された前記連続系方程式の記述に基づいて、第 1 のプログラムを生成する第 1 のプログラム生成ステップと、

前記解析ステップにおいて抽出された前記状態遷移に伴う連続系方程式の切り替えに関する記述に基づいて、第 2 のプログラムを生成する第 2 のプログラム生成ステップと、

前記解析ステップにおいて抽出された前記連続系方程式で表現される処理以外の処理に関する記述に基づいて、第 3 のプログラムを生成する第 3 のプログラム生成ステップと、

前記第 1 のプログラムを実行することにより、前記連続系方程式をシミュレーション実行可能なデータ構造に変換する変換ステップと、

前記第 2 のプログラムを実行することにより、イベントの発生に応じて前記連続系方程式の有効・無効を切り替える切り替えステップと、

前記切り替えステップにおいて有効とされた前記連続系方程式に対応する前記データ構造を用いて、時間軸に沿った数値積分により該連続系方程式を解いて、前記機構の挙動を表すデータを出力するシミュレーションを実行するシミュレーション実行ステップと、

前記第 3 のプログラムを実行することにより、前記連続系方程式で表現される処理以外の処理を実行する処理実行ステップと
を具備することを特徴とするシミュレーション方法。

【請求項 2】 前記イベントは、前記機構を制御する機構制御ソフトウェアを含

む外部からの制御信号を含むことを特徴とする請求項 1 に記載のシミュレーション方法。

【請求項 3】 前記イベントは、前記機構の内部変数の評価結果を含むことを特徴とする請求項 1 に記載のシミュレーション方法。

【請求項 4】 ハイブリッドモデルを利用した記述データに基づいて、対象とする機構の挙動を時間軸に沿ってシミュレーションする手順をコンピュータに実行させるシミュレーションプログラムであって、

前記記述データに含まれる連続系方程式の記述、状態遷移に伴う該連続系方程式の切り替えに関する記述、及び前記連続系方程式で表現される処理以外の処理に関する記述を抽出するため、前記記述データを解析する解析手順と、

前記解析ステップにおいて抽出された前記連続系方程式の記述に基づいて、第 1 のプログラムを生成する第 1 のプログラム生成手順と、

前記解析ステップにおいて抽出された前記状態遷移に伴う連続系方程式の切り替えに関する記述に基づいて、第 2 のプログラムを生成する第 2 のプログラム生成手順と、

前記解析ステップにおいて抽出された前記連続系方程式で表現される処理以外の処理に関する記述に基づいて、第 3 のプログラムを生成する第 3 のプログラム生成手順と、

前記第 1 のプログラムを実行することにより、前記連続系方程式をシミュレーション実行可能なデータ構造に変換する変換手順と、

前記第 2 のプログラムを実行することにより、イベントの発生に応じて前記連続系方程式の有効・無効を切り替える切り替え手順と、

前記切り替えステップにおいて有効とされた前記連続系方程式に対応する前記データ構造を用いて、時間軸に沿った数値積分により該連続系方程式を解いて、前記機構の挙動を表すデータを出力するシミュレーションを実行するシミュレーション実行手順と、

前記第 3 のプログラムを実行することにより、前記連続系方程式で表現される処理以外の処理を実行する処理実行手順と
をコンピュータに実行させることを特徴とするシミュレーションプログラム。

【発明の詳細な説明】**【0001】****【発明の属する技術分野】**

本発明は、コンピュータを用いて機械等の挙動のシミュレーションを行うための方法およびプログラムに関し、特に本発明はハイブリッドモデルを用いるものである。

【0002】**【従来の技術】**

現在、コンピュータを用いて機械やプラント等の挙動のシミュレーションを行う際に、ハイブリッドモデリングと呼ばれる手法が使われている。ハイブリッドモデルを用いたシミュレーションは「ハイブリッドシミュレーション」と呼ばれている。このようなシミュレーション挙動をするシステムを「ハイブリッドシステム」と呼ぶこともある。

【0003】

シミュレーションの目的で作成されるハイブリッドモデルは、概念的には常微分方程式や代数方程式を連立させた連立方程式によって表現される連続系モデルと、イベント発生に伴う状態遷移を表現するための状態遷移モデルとを組み合わせたモデルである。ハイブリッドモデルによれば、連続系モデルで表現される状態が外部からのイベントなどにより瞬時に切り替わるシステムを表現することができる。

【0004】

ハイブリッドモデルを記述するための言語として、米国ゼロックス社（商標）のパロアルト研究所にて創作されたHCC (Hybrid Concurrent Constraint Programming) と呼ばれる言語がある（下記非特許文献1参照）。HCCは発展途上にあり、現在も米国NASAのエイムズ研究所において研究が進められている。HCCは制約処理プログラミング（コンストレイントプログラミング）と呼ばれる技術の一種であり、連続系モデルを表現する常微分方程式や代数方程式を制約として扱い、これら方程式をそのまま順不同で記述することができる。このような制約記述に、状態遷移を制御する記述を付加してHCC言語のハイブリッドモデ

ルは完成される。HCCによれば、方程式をそのまま制約として羅列する（プログラミングする）ことができ、複雑なモデルを記述可能である。

【0005】

このようにハイブリッドモデルの技術を用いれば、系の特性を常微分方程式などでモデル表現し、初期状態から時間の推移に従ってどのような挙動を示すかをシミュレーションすることができる。

【0006】

微分方程式などで表現可能な対象や現象を的確にモデル化できるハイブリッドモデルの技術の応用例として、ソフトウェアにより機構が制御されるメカトロニクス機器の機構シミュレーションがある。かかる機構シミュレーションによれば、機構の実機が存在しない状況においても、当該機構を制御する制御ソフトウェアのプロトタイピング、テスト、あるいはデバッグなどが行えるようになる。

【0007】

【非特許文献1】

インターネット

<URL : <http://www2.parc.com/spl/projects/mbc/publications.html#cclanguages>>

【0008】

【発明が解決しようとする課題】

しかし、ハイブリッドモデルを扱うことのできる公知のプログラム言語は、必ずしもメカトロニクス機器の機構シミュレーションに応用することを目的に開発されたものではなく、このため次のような問題点がある。

【0009】

例えば米国ゼロックス社（商標）のHCCは、インタプリタ型のプログラミング言語であり、たとえば制御ソフトウェアが機構に送信するアクチュエータへの動作コマンドなどを、シミュレータの外部から制御信号として受け取るような構成とする場合、外部関数などを個別に定義する必要がある上、プログラミングにおいてかなりの工夫を必要とする。

【0010】

本発明はかかる事情を考慮してなされたものであり、ハイブリッドモデルを用いて複雑な機構系を簡便かつ正確にモデル化でき、該機構系を制御する制御ソフトウェアとの連携シミュレーションにも好適なシミュレーション方法およびプログラムを提供することを目的とする。

また、外部との連携などに関わる処理を記述するプログラムを直接ハイブリッドモデル記述の一部に含めることを可能にし、シミュレーションの適用範囲を拡大することを目的とする。

【0011】

【課題を解決するための手段】

上記課題を解決し目的を達成するために、本発明では、ハイブリッドモデルを利用した記述データに基づいて、対象とする機構の挙動を時間軸に沿ってシミュレーションするシミュレーション方法であって、前記記述データに含まれる連続系方程式の記述、状態遷移に伴う該連続系方程式の切り替えに関する記述、及び前記連続系方程式で表現される処理以外の処理に関する記述を抽出するため、前記記述データを解析する解析ステップと、前記解析ステップにおいて抽出された前記連続系方程式の記述に基づいて、第1のプログラムを生成する第1のプログラム生成ステップと、前記解析ステップにおいて抽出された前記状態遷移に伴う連続系方程式の切り替えに関する記述に基づいて、第2のプログラムを生成する第2のプログラム生成ステップと、前記解析ステップにおいて抽出された前記連続系方程式で表現される処理以外の処理に関する記述に基づいて、第3のプログラムを生成する第3のプログラム生成ステップと、前記第1のプログラムを実行することにより、前記連続系方程式をシミュレーション実行可能なデータ構造に変換する変換ステップと、前記第2のプログラムを実行することにより、イベントの発生に応じて前記連続系方程式の有効・無効を切り替える切り替えステップと、前記切り替えステップにおいて有効とされた前記連続系方程式に対応する前記データ構造を用いて、時間軸に沿った数値積分により該連続系方程式を解いて、前記機構の挙動を表すデータを出力するシミュレーションを実行するシミュレーション実行ステップと、前記第3のプログラムを実行することにより、前記連続系方程式で表現される処理以外の処理を実行する処理実行ステップとを具備す

ることを特徴とするシミュレーション方法を提供する。

【0012】

また、本発明では、ハイブリッドモデルを利用した記述データに基づいて、対象とする機構の挙動を時間軸に沿ってシミュレーションする手順をコンピュータに実行させるシミュレーションプログラムであって、前記記述データに含まれる連続系方程式の記述、状態遷移に伴う該連続系方程式の切り替えに関する記述、及び前記連続系方程式で表現される処理以外の処理に関する記述を抽出するため、前記記述データを解析する解析手順と、前記解析ステップにおいて抽出された前記連続系方程式の記述に基づいて、第1のプログラムを生成する第1のプログラム生成手順と、前記解析ステップにおいて抽出された前記状態遷移に伴う連続系方程式の切り替えに関する記述に基づいて、第2のプログラムを生成する第2のプログラム生成手順と、前記解析ステップにおいて抽出された前記連続系方程式で表現される処理以外の処理に関する記述に基づいて、第3のプログラムを生成する第3のプログラム生成手順と、前記第1のプログラムを実行することにより、前記連続系方程式をシミュレーション実行可能なデータ構造に変換する変換手順と、前記第2のプログラムを実行することにより、イベントの発生に応じて前記連続系方程式の有効・無効を切り替える切り替え手順と、前記切り替えステップにおいて有効とされた前記連続系方程式に対応する前記データ構造を用いて、時間軸に沿った数値積分により該連続系方程式を解いて、前記機構の挙動を表すデータを出力するシミュレーションを実行するシミュレーション実行手順と、前記第3のプログラムを実行することにより、前記連続系方程式で表現される処理以外の処理を実行する処理実行手順とをコンピュータに実行させることを特徴とするシミュレーションプログラムを併せて提供する。

【0013】

【発明の実施の形態】

以下、図面を参照しながら本発明の実施形態を説明する。

【0014】

図1は本発明の実施形態に係る機構シミュレータの概略構成を示すブロック図である。

【0015】

本実施形態は、ハイブリッドモデル前処理部201と、ハイブリッドモデルシミュレーション実行部102とにより構成されている。ハイブリッドモデル記述104は、ハイブリッドモデル記述言語等で記述されたソースプログラムであって、本実施形態に係るハイブリッドモデル前処理部201への入力である。制御信号106はハイブリッドモデルシミュレーション実行部102への入力であり、この制御信号106は後述する機構制御ソフトウェアもしくは機構制御ソフトウェアのシミュレータから与えられる。また、本実施形態に係る機構ハイブリッドモデルシミュレーション実行部102からの出力は、シミュレーション結果としての変数値の演算結果およびその時間履歴であり、変数値時間履歴記憶部105に対して出力される。

【0016】

図1に示すように、ハイブリッドモデル前処理部201は、制御情報解析部110を備える。また、ハイブリッドモデルシミュレーション実行部102は、イベント処理部111、方程式構文解析部112、方程式データ記憶部114、連続系方程式切り替え部115、付加処理実行部205と、及び連続系シミュレーション部103とを備える。なお、本実施形態は一般的なコンピュータを用いて構成することができ、その基本的なハードウェア構成として、図示しないが中央演算ユニット（CPU）、メモリ、外部記録装置、通信インターフェース（I/F）、および表示装置やキーボード、マウス等の入力装置を備える。また、これらのハードウェアを制御するためのオペレーティングシステム（OS）を備える。また、本発明の実施形態に係る機構シミュレータは、このようなオペレーティングシステム上で動作するアプリケーションソフトウェアとして実装することができる。

【0017】

このような本実施形態に係る機構シミュレータの構成及びその処理手順について説明する前に、ここではまずハイブリッドモデル記述104がどのように記述されるのか、具体例を挙げて説明する。

【0018】

図2および図3は、具体例に係るハイブリッドモデルの記述対象である機構を示す図であって、この機構は、バルブ301、バネ303、およびピストン302を備えた簡素な構造を有するシリンダ装置である。

【0019】

バルブ301は外部からの指令（イベント）に応じて開閉動作する。これによりシリンダ装置内の空気の流れを図2のように右側に変更するイベントを以下、「Right」と呼び、空気の流れを図3のように左側に変更するイベントを「Left」と呼ぶ。図2は、バルブ301にRightのイベントが与えられた状態を示しており、ピストン302には紙面左向きへの力が作用している。この状態を示す運動方程式はシリンダ装置の下部に示してあるように、「 $-F = m x''$ 」である。これに対し図3は、バルブ301にLeftのイベントが与えられた状態を示しており、空気の流れの向きが変わり、運動方程式は同図のように「 $F = m x''$ 」に変化している。

【0020】

図4は、このような機構の挙動を、状態変化とそれぞれの状態に対応する運動方程式からなる状態遷移図として表現したものである。ハイブリッドモデルは、この図4に示されるような状態遷移と、各状態の記述が微分方程式や代数方程式あるいはそれらの連立方程式（連続系方程式）で表現されるものを指す。図4によれば、状態が2つあって、かかる2つの状態間に状態遷移が存在することがわかる。

【0021】

本発明では、図4の状態遷移図をもとに具体的なハイブリッドモデルの内容をHCC (Hybrid Concurrent Constraint Programming) 言語で記述するとともに、連続系方程式で表現される処理以外の処理、例えば、外部との連携などに関わる処理（以下「付加処理」という）を所定のプログラム言語で記述することによりハイブリッドモデル記述104が生成される。

【0022】

図5は、本発明にかかるハイブリッドモデル記述に相当するプログラムの一例を示す図である。図5において、（ソース）プログラムの論理行番号を仮にL1

～L 2 3とする。L 3、L 4、およびL 8は、上記した機械装置の初期状態やバルブ操作タイミング等の運転条件の記述に相当し、L 5およびL 6は図4に示した状態遷移の表現記述に相当する。また、L 9～L 2 3は付加処理に係る記述であり、ここでは「現在の状態をファイルに書き込む」という処理を示している。なお、L 9～L 1 7はC言語によるプログラム記述となっている。

【0023】

HCC言語では、図から分かるように、運動方程式をプログラム内でそのまま記述することができる。また、それぞれの状態へ遷移する条件は、「always if」に続いて記述し、それぞれの状態から遷移していく条件は「watching」に続けて記述される。

【0024】

なお、HCCでは、プログラムの記述の順序（例えば図5における論理行番号L 1→L 8の順序）にそって実行されるわけではない。HCCでは、個別のプログラム記述のうち、シミュレーションを実行する時間軸に沿って成立するものが探索され、実行される。すなわち、論理行番号L 1→L 8の順序は、実行順序とは関係がない。たとえば、シミュレーションを開始した時点では、L 3およびL 8のみが有効である。ここで、イベントRight (e v 1) がL 3により発生するため、L 6の前提条件であるRightが有効となり、L 6に記述されている運動方程式e q 2が有効になる。つまり、図4の左側の状態からシミュレーションが実行されることになる。

【0025】

さらに、時間が50になるとL 4が有効となり、イベントLeft (e v 2) が発生し、L 6の遷移条件（「watching」以下、すなわちLeft）が有効となって、L 6の運動方程式e q 2が無効となる。これに代わって、L 5の前提条件が有効となり、運動方程式e q 1が有効となる。

【0026】

なお、以上のようなプログラム例は外部からのイベント (e v 3, e v 4) によって状態が遷移する場合を記述したものであったが、勿論、内部の状況によって状態を変化させてもよい。たとえば、図2においてバルブ301が切り替えら

れない場合には、移動するピストン 302 がバネ 303 に接触し、該バネ 303 からの反力を受けるようになる。すなわちピストン 302 の位置に関して、外部からのイベントが無い場合でも状態遷移が起こる場合が存在する。このような場合は、例えば x が正であるかどうかといった内部変数の評価式（不等式）による評価結果に基づいて状態遷移の必要性を判断できる。

【0027】

一般的に、ハイブリッドモデルは、常微分方程式や代数方程式を連立させた連立方程式によって表現される連続系モデルと、イベント発生に伴う状態遷移を表現するための状態遷移モデルとを組み合わせたモデルである。ハイブリッドモデルによれば、連続系モデルで表現される状態が外部からのイベントなどにより瞬時に切り替わるシステムを表現することができる。

【0028】

さらに、時間が 100 になると L19 が有効となり、イベント E (ev5) が発生し、cPrint という関数が引数 x の値とともに呼び出される。かかる処理は、その時点での状態をファイルに保存する付加処理が実行されることを示しており、かかる処理の具体的なメカニズムは、L9～L17 において C 言語で記述されている。

【0029】

次に、ハイブリッドモデル前処理部 201 における処理について説明する。ハイブリッドモデル記述 104 は、まずハイブリッドモデル前処理部 201 の制御情報解析部 110 において処理され、モデル方程式登録プログラム 202、イベント制御プログラム 203、及びシミュレーション実行時付加処理プログラム 204 が生成される。また、ハイブリッドモデルシミュレーション実行部 102 を構成するソフトウェアモジュールとして、モデル方程式の登録を行うための関数及び連続系方程式を切り替えるための関数が A P I (Application Program Interface) 関数として提供される。モデル方程式登録プログラム 202 およびイベント制御プログラム 203 は、該当する上記 A P I 関数を呼び出す記述を、入力されたハイブリッドモデル記述 104 に沿って適切に組み合わせたプログラムである。この観点から考えると、ハイブリッドモデル前処理部 201 は、入力をハ

イブリッドモデル記述 104 とし、出力を例えば C 言語の API 関数呼び出しの記述を含む C プログラム (ソース) とするような、一種のコンパイラと考えることもできる。このようなモデル方程式登録プログラム 202 とモデル方程式制御プログラム 203 は、さらに C 言語などのコンパイラによりコンパイルされ、例えば実行時に動的にリンク可能なライブラリが生成される。ハイブリッドモデルシミュレーション実行部 102 は、シミュレーション実行にあたって、生成された動的リンクライブラリがリンクされ、入力ハイブリッドモデルを忠実に再現するシミュレーションプログラムが完成し、実行可能になる。その実行時には、まず方程式構文解析部 112 を起動する API 関数が呼ばれ、その後に連続系方程式切り替えの API 関数群が実行されて連続系シミュレーションが遂行される。

【0030】

ハイブリッドモデルシミュレーション部 102 のアプリケーションインターフェースを構成する具体的なソフトウェアモジュールの仕様などは様々考えられるが、ここでは説明の都合上、以下の 3 つの API 関数が最低定義されているとする。なお、プログラミング言語は C 言語とする。

【0031】

【数 1】

```
int XXX_AddEqnData(char *eqn, int *err)
int XXX_ActivateEqn(int eqnid)
int XXX_DeActivateEqn(int eqnid)
```

【0032】

1 つ目の API 関数 XXX_AddEqnData は、1 つの連続系方程式を表す文字列のポインタを引数に指定する。XXX_AddEqnData は、この連続系方程式を構文解析し、連続系方程式の記述をシミュレーション実行可能なデータ構造 (内部データ表現) に変換し、かかる内部データ表現を方程式データ記憶部 114 に登録する処理を行う。なお、この連続系方程式には、ユニークな ID 番号が割り当てられる。

【0033】

たとえば「 $ab/\cos(a-(c+b))-3c$ 」という式が与えられたと仮定すると、上記内部データ表現として図 6 のような木構造を生成する。この木構造において、例え

ば参照数字 61 は線形多項式の親ノード（節）、62 は掛け算のノード、63 は割り算のノード、64 は外部関数（四則演算以外の意）のノード、65 は線形多項式を構成する各項のノードを表している。本例において、木構造の葉に相当するものはすべて変数 (a, b, c) であり、これらに実数の係数が加わって線形式となる。線形式は cos などの外部関数の引数になったり、掛け算や割り算の対象となる。変数には、別途、値が確定しているかどうかのフラグが設けられており、またこのような木構造のデータに基づいて該変数の現在の値が保持される。木構造のすべての葉の値（すなわち変数の値）が確定していれば、式の値を計算することができる。方程式データ記憶部 114 では、式の値の計算などを高速に行うことができるように、予め内部のデータ構造をつなぎ合わせて木構造を構成してある。

上記処理において何らかのエラーが発生した場合には、err にエラーコードがセットされる。正常に処理が終了した場合は、登録された方程式の ID 番号を返り値とする。

【0034】

2 番目の API 関数 XXX_ActivateEqn は、引数に指定された方程式の ID 番号に相当する方程式を有効にする。もし、すでに有効となっている方程式が指定されている場合には何もしない。返り値はエラーコードである。

【0035】

3 番目の API 関数 XXX_DeActivateEqn は、XXX_ActivateEqn とは逆に、引数に指定された方程式の ID 番号に相当する方程式を無効にする。すでに無効となっている方程式が指定された場合には何もしない。

制御情報解析部 110 は、まず XXX_AddEqnData を必要な方程式について順に呼ぶ関数 (InitEqnData) を生成する。これがモデル方程式登録プログラム 202（第 1 のプログラム）に相当する。

【0036】

また、制御情報解析部 110 は、シミュレーション実行の際に、時間が Δt 進むごとに条件のチェックおよび方程式の変更（入れ替え）を行う関数 (ChangeEqn) も生成する。これはイベント制御プログラム 203（第 2 のプログラム）に

相当する。

【 0 0 3 7 】

さらに、制御情報解析部 1 1 0 は、付加処理を実行するための記述をハイブリッドモデル記述 1 0 4 から抽出する。かかる記述が、シミュレーション実行部付加処理プログラム 2 0 4（第 3 のプログラム）に相当する。シミュレーション実行部付加処理プログラム 2 0 4 はハイブリッドモデル記述 1 0 4 中でもともと C 言語などにより記述されたプログラムであり、例えば図 5 に示したハイブリッドモデル記述 1 0 4 中では、module 文が参照され、C 言語で記述された部分の抽出が行なわれる。抽出されたシミュレーション実行部付加処理プログラム 2 0 4 は、C 言語コンパイラによりコンパイルされ、動的にリンク可能なライブラリが生成される。付加処理実行部 2 0 5 は、この動的にリンク可能なライブラリを呼び出すインターフェースの役割を担う。

【 0 0 3 8 】

上記したようなハイブリッドモデル前処理部 2 0 1 における処理により、例えば、図 5 に示したハイブリッドモデル記述について、以下のような C 言語のソースプログラムが自動生成される。

【 0 0 3 9 】

【数 2】

```
static char eqn1[] = "f = m x' ' " ;
static char eqn2[] = "-f = m x' ' " ;
static int eqn1id;
static int eqn2id;
int InitEqnData0
{
    int err;
    eqn1id = XXX_AddEqnData(eqn1,&err);
    if(err != 0) return err;
    eqn2id = XXX_AddEqnData(eqn2,&err);
    if(err != 0) return err;
    return 0;
}

int ChangeEqn0
{
    int err;
    BOOL GetEvent(char *eventname);
    if( GetEvent(Left) ){
        err = XXX_ActivateEqn(eqn1id);
        if( err != 0 ) return err;
        XXX_DeActivateEqn(eqn2id);
        if( err != 0 ) return err;
    }
    if( GetEvent(Right) ){
        XXX_ActivateEqn(eqn2id);
        if( err != 0 ) return err;
        XXX_DeActivateEqn(eqn1id);
        if( err != 0 ) return err;
    }
    if( GetEvent(E) ){
        cPrint(x);
    }
    return 0;
}

int cPrint(int num)
{
    FILE* fp = fopen( "hoge.txt" , "a" );
    fprintf(fp," %d" ,num);
    fclose(fp);
}
```

【 0 0 4 0】

なお、GetEventは、名前で指定されたイベントが生起しているかどうかをチェ

ックする関数である。

以上のプログラムは、上述したようにC言語コンパイラによってコンパイルされ、さらに動的リンクライブラリの形式に整えられ、実行時にリンクされる。

【0041】

なお、本実施形態では、プログラム言語としてC言語を用いた例について説明したが、本発明はこれに限定されるものではなく、例えばC++言語、Spec-C言語等の他のプログラム言語を用いてもよい。

【0042】

次に、シミュレーションの実行について説明する。シミュレーション実行時には、ハイブリッドモデルシミュレーション実行部102が起動され、機構制御ソフトウェアシミュレータ108などから得られる制御信号106を受信しながら、連続系方程式の値を計算することでシミュレーション実行が行われる。このとき、連続系方程式切り替え部115は、上記したイベント制御プログラム203に基づいてイベント処理部111からの指令により、連続系方程式の切り替えを有効・無効のフラグを用いて実行する。図2の状態では、図5の運動方程式eq1は無効であり、運動方程式eq2が有効になっている。ここで、Leftのイベントが発生した図3の状況においては、図5の運動方程式eq1を有効にし、運動方程式eq2を無効にするようフラグを操作する。これら有効・無効のフラグは方程式データ記憶部114に記憶される方程式それぞれの属性データとして管理される。

【0043】

連続系シミュレーション部103は、方程式データ記憶部114を参照し、同記憶部114に木構造の形式で格納されている連続系方程式の内部データ表現を演算対象として、時間ステップづつ数値積分を実行する。シミュレーションは、常微分方程式及び代数多項式の連立からなる非線形連立方程式についての初期値問題である。このため、例えば図2に示される初期状態が与えられている。具体的には、例えば一般によく使われているルンゲクッタアルゴリズムを用いて解の値を計算する。

【0044】

必要なデータは機構シミュレータから出力を行い、さらに連続系方程式切り替え部 115 の処理に戻り、上記の処理を繰り返すことにより必要な時間のシミュレーションを実行する。シミュレーション結果は、変数値時間履歴記憶部 105 に保存され、シミュレーション終了後の分析などに利用される。

【0045】

図 7 は、以上説明した本発明の第 1 実施形態に係るシミュレーションにおける一連の処理手順を示すフローチャートである。

【0046】

まず、図 1 に示すように、ハイブリッドモデル記述 104 を制御情報解析部 110 に入力し、ハイブリッドモデルの構文解析を行い（ステップ 501）、上記したモデル方程式登録プログラム 202、イベント制御プログラム 203、及びシミュレーション実行時付加処理プログラム 204 を生成する。かかるプログラムのうち、モデル方程式登録プログラム 202 により、連続系方程式をシミュレーション実行可能なデータ構造に変換するとともに方程式データ記憶部 114 に登録する（ステップ 502）。

【0047】

以上でシミュレーション実行のための前処理を終え、ここからシミュレーション実行の段階に移る。まずは機構制御ソフトウェアシミュレータ 108 等から与えられる制御信号 106 の受信処理を行う（ステップ 503）。ここで、必要であれば、受信した制御信号に基づく値を適宜、変数に代入する。次に、状態変化の必要有無をステップ 504 にて判定する。状態変化が必要な場合、連続系方程式切り替え部 115 は、該当する連続系方程式について、その有効・無効のフラグを操作することにより、必要に応じて連続系方程式の切り替えを行う（ステップ 505）。

【0048】

次に、連続系シミュレーション部 103 は数値積分を実行する（ステップ 506）。

【0049】

その後、付加処理を実行する（ステップ 511）。具体的な付加処理の例とし

ては、画面上に処理の進捗状況を表示したり、シミュレーションに関わるデータをファイルに出力するなどの処理が考えられる。

【0050】

次に、機構制御ソフトウェアシミュレータ108に対し機構シミュレータ側からセンサー情報などが送信される（ステップ508）。さらに、ステップ509にて終了判断を行ったのち、所定の終了条件が成立するまで時間を1ステップ進め（ステップ510）、ステップ503以降の処理手順を繰り返すことによりシミュレーション実行が行われる。

【0051】

以上の処理を図5に示したハイブリッドモデル記述について実行した場合の時系列的な流れを図8に示す。t=0は初期状態を示し、t=0でイベントLeftが発生し連続系方程式が切り替えられ、t=100でイベントEが発生し付加処理が実行される。かかるイベント間は連続系のシミュレーションが実行されている。

【0052】

ハイブリッドモデルシミュレーション実行部102が機構制御ソフトウェアシミュレータと連携し、全体として効率的にシミュレーションが実行されるためには、専らハイブリッドモデル記述104の内容に応じた処理を行う504、505、および506の各ステップと、専らデータ送受信等を行う503、507、および508の各ステップとの間の実行順序が適切かつ効率的に制御されなければならない。ここで、既に述べたように従来では、ハイブリッドモデル記述言語（例えばHCC言語）の特性（どの行が最初に実行されるか分からない）に起因し、上記実行順序を適切かつ効率的に制御するようプログラミングするには、極めて高度なテクニックを要し、難易度の高い作業である。また、単なる変数の値を扱うのではなく、外部から得られた情報を変数に代入してからハイブリッドモデルの処理を行うなど、外部とのインターフェースに関わる特別な外部関数などを用意し、これらをモデルの作成者が記述しなければならなかった。

【0053】

これに対し、本発明の実施形態によれば、ハイブリッドモデルシミュレーショ

ン実行部 102 が図 7 に示した一連の処理を実行するよう構成されていることから、上記のような問題を回避できる。すなわち、機構制御ソフトウェアシミュレータ 108 等の外部（プロセス）との間の通信に相当するソフトウェア部分は、ハイブリッドモデルの中に、C 言語などのプログラム言語として記述でき、その機能が自動的にシミュレーションの実行に組み込まれる。これにより、複雑なシミュレーションの手順も柔軟にモデル化およびプログラミングが可能になる。

【0054】

【発明の効果】

以上説明したように、本発明によれば、ハイブリッドモデルを用いて複雑な機構系を簡便かつ正確にモデル化でき、該機構系を制御する制御ソフトウェアとの連携シミュレーションにも好適なシミュレーション方法およびプログラムを提供できる。

【図面の簡単な説明】

【図 1】 本発明の第 1 実施形態に係る機構シミュレータの概略構成を示すブロック図。

【図 2】 ハイブリッドモデル記述を説明するための具体例に係るシリンダ装置のある状態を示す図。

【図 3】 ハイブリッドモデル記述を説明するための具体例に係るシリンダ装置の別の状態を示す図。

【図 4】 ハイブリッドモデル記述を説明するための具体例に係るシリンダ装置の状態遷移を示す図。

【図 5】 ハイブリッドモデル記述の内容を示す図。

【図 6】 1 つの連続系方程式を構文解析した結果得られる内部データ表現の説明図。

【図 7】 機構シミュレーションの処理手順を示すフローチャート。

【図 8】 機構シミュレーションの時系列的な流れの説明図。

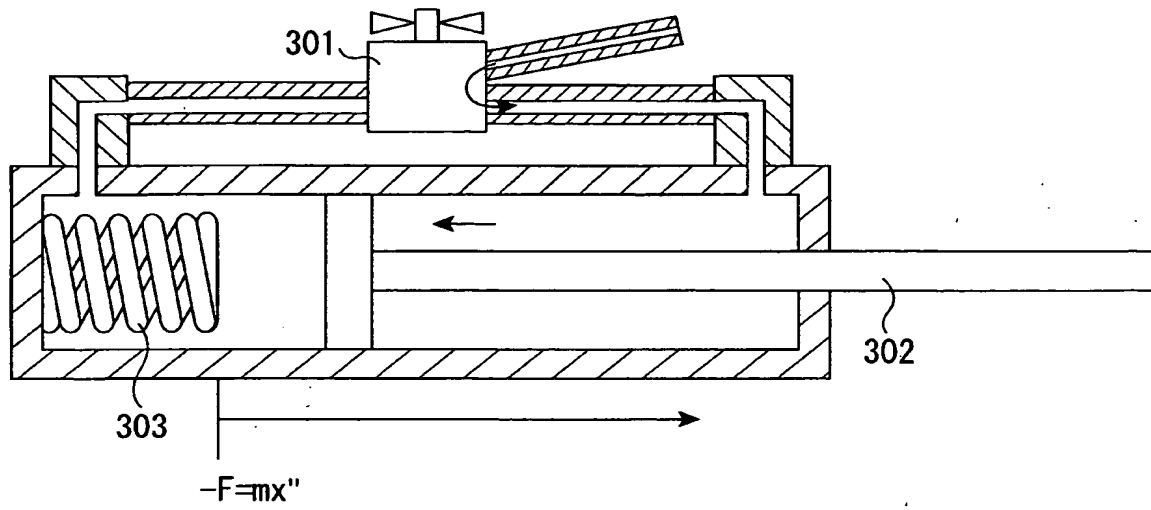
【符号の説明】

102 ハイブリッドシミュレーション実行部

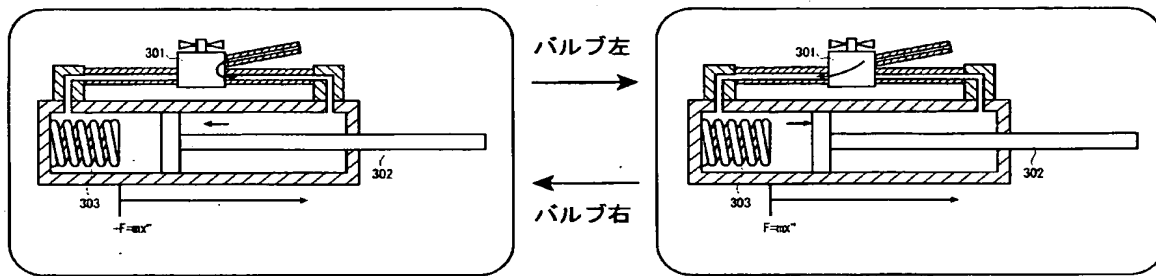
103 連続系シミュレーション部

- 104 ハイブリッドモデル記述
- 105 変数値時間履歴記憶部
- 106 制御信号
- 110 制御情報解析部
- 111 イベント処理部
- 112 方程式構文解析部
- 114 方程式データ記憶部
- 115 連続系方程式切り替え部
- 201 ハイブリッドモデル前処理部
- 202 モデル方程式登録プログラム
- 203 イベント制御プログラム
- 204 シミュレーション実行時付加処理プログラム
- 205 付加処理実行部

【図 3】



【図 4】



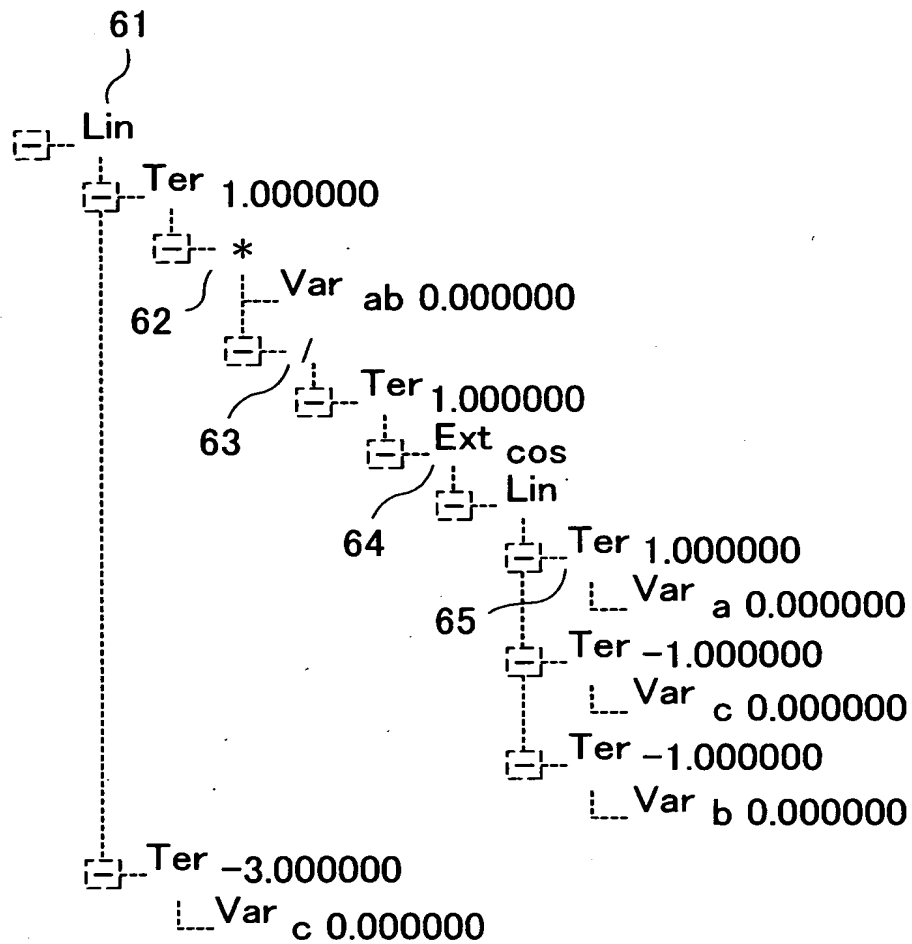
【図 5】

```

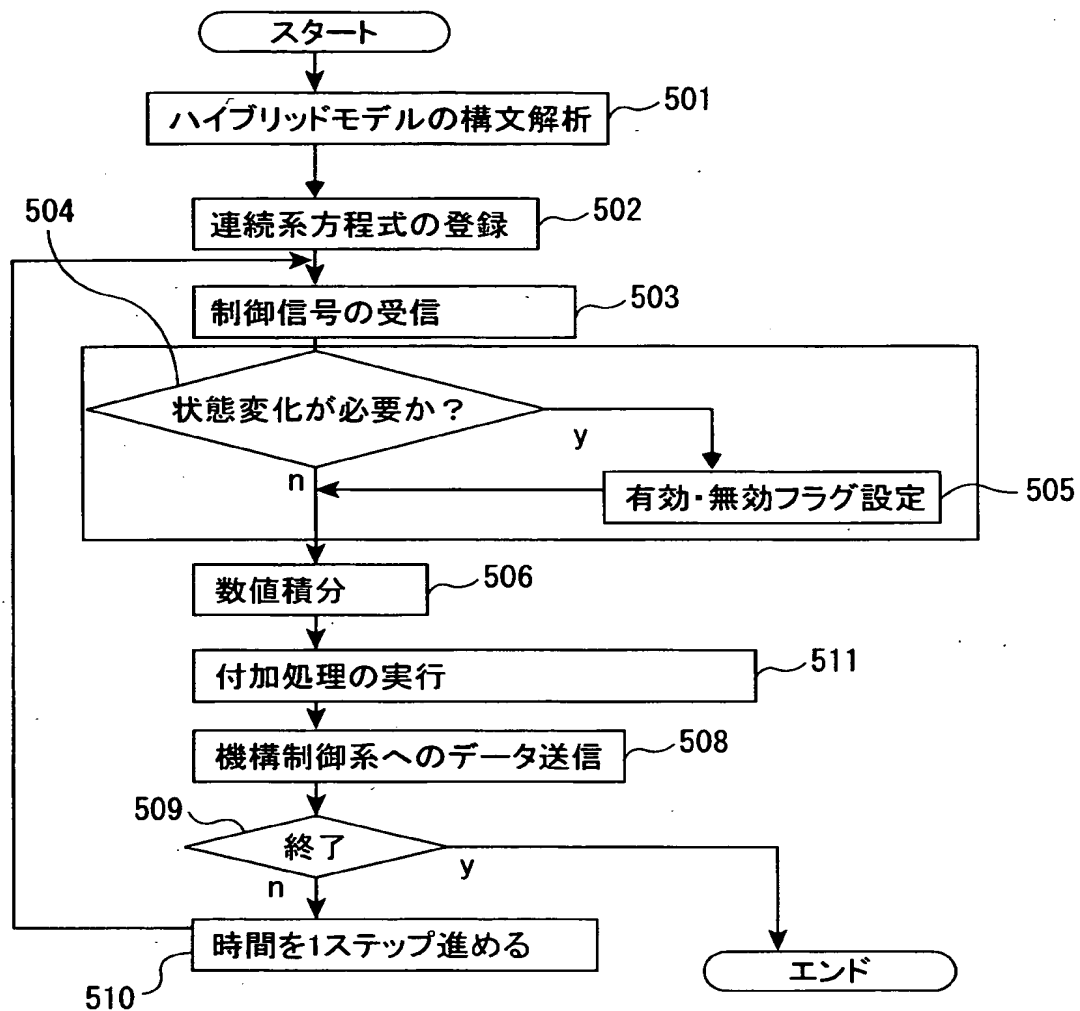
L1  #define m 1
L2  #define f 100
L3  Right ~ ev1
L4  wait 50 do Left ~ ev2
L5  always if Left then do always F = m * x ~ ev3 watching Right,
L6  always if Right then do always -F = m * x ~ eq1 watching Left,
L7  sample(x), ~ ev4
L8  x = 0, x' = 0, ~ eq2
L9  module "C"
L10 {
L11   int cPrint (int num)
L12   {
L13     FILE * fp = fopen( "hoge.txt", "a" );
L14     fprintf (fp, "%d", num)
L15     fclose (fp)
L16   }
L17 }
L18 int cPrint (int num)
L19 wait 100 do E ~ ev5
L20 process (E) {
L21   cPrint (a);
L23 }

```

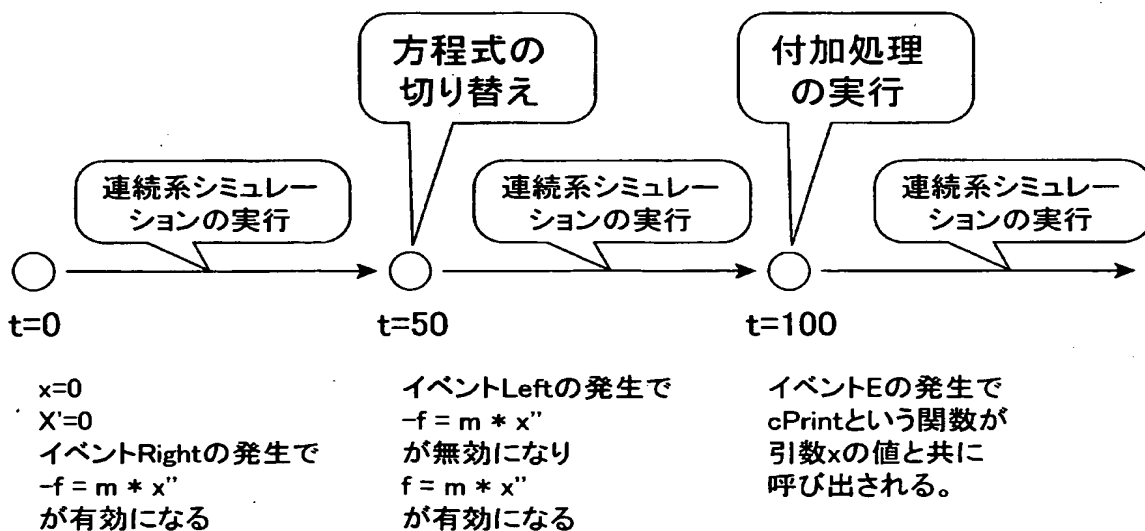
【図 6】



【図 7】



【図 8】



【書類名】 要約書

【要約】

【課題】 機構系を制御する制御ソフトウェアとの連携シミュレーションにも好適なシミュレーション方法およびプログラムを提供。

【解決手段】 ハイブリッドモデルを利用した記述データに含まれる連続系方程式の記述、状態遷移に伴う該連続系方程式の切り替えに関する記述、及び前記連続系方程式で表現される処理以外の処理に関する記述を抽出するため、前記記述データを解析する解析ステップ501と、前記解析結果に基づいて、第1乃至第3のプログラムを生成するプログラム生成ステップと、前記第1乃至第3のプログラムを実行することにより、前記機構の挙動を表すデータを出力するシミュレーションを実行するシミュレーション実行ステップ505、506及び前記連続系方程式で表現される処理以外の処理を実行する処理実行ステップ511とを具備することを特徴とするシミュレーション方法およびプログラム。

【選択図】 図7

認定・付加情報

特許出願の番号	特願 2002-376210
受付番号	50201970910
書類名	特許願
担当官	第七担当上席 0096
作成日	平成15年 1月 6日

<認定情報・付加情報>

【提出日】 平成14年12月26日

次頁無

特願 2002-376210

出願人履歴情報

識別番号

[000003078]

1. 変更年月日 2001年 7月 2日
[変更理由] 住所変更
住 所 東京都港区芝浦一丁目1番1号
氏 名 株式会社東芝
2. 変更年月日 2003年 5月 9日
[変更理由] 名称変更
住所変更
住 所 東京都港区芝浦一丁目1番1号
氏 名 株式会社東芝